Software Test & Performance CONFERENCE

Performance. Agility. Collaboration.
October 19–23, 2009

Data patterns in performance testing

Ross Collard
Principal Consultant
Collard and Company
www.rosscollard.com

Dan Downing
Principal Consultant
MENTORA GROUP
www.mentora.com

Objectives

- **Raise your performance testing *Data IQ***
- **Target audience**
  - Performance testers
- **Acknowledgements:**
  - The attendees of WOPR12* contributed content to this presentation:

*** Workshop on Performance and Reliability**

Henry Amistadi	Charlie Audritsh	Basim Baassiri	
Andy Hohenner	Paul Holland	Karen N. Johnson	Mike Kelly
Yury Makedonov	Jeff Pickett	Alex Podelko	Eric Proegler
Raymond Rivest	Roland Stens	Nick Wolf	

What is Live Data?

- **Data captured from live operations; can encompass**
  - Source data (data files / user inputs to apps)
  - Logs of workflows/click-streams
  - Transaction data extracted from databases
  - Atomic event data (e.g., the time an event occurred)
  - Data derived from atomic data (e.g., response time between events)
  - Metadata (error rates, completed transactions)

Using live data in a load test is good, but often does not yield reliable performance predictions

Current Best Practices

- Most testers have superficial understanding of live data
- Live data selection is informal
- Developing profiles of users & usage reduces risk but is not sufficient
- Some testers enhance live data, but lack effective techniques
- Statistical validation of live data rarely employed
- Related issues often neglected, such as privacy & confidentiality

Live data should be captured and enhanced based on context and guided by specification & design

Selecting & Enhancing Live Data

- **What live data to use?**
  - As cannot test everything, focus on data that links to the biggest payoff opportunities
  - Look into the details of specific use cases, known problems, stakeholder concerns,
- How to capture, enhance?
  - Understand data pattern categories
  - Understand capture options available in your environment
  - Choose enhancement pattern(s) based on test objectives and context guideposts

Test Data Pattern Categories

1. **Mainstream –** Patterns unlikely to be enhanced
   - Baseline, SLA Compliance, Background noise, normal peak…
2. **Mainstream –** Patterns with most common enhancements
   - Pristine, grown, peak-peak, privacy-protected, SLA violation…
3. **Measurable Behavior –** Most commonly engineered
   - Response Time, Throughput, resource utilization, testability,…
4. **Stress Patterns –** Data replay at higher rates
   - Zero think-time or super-user, shared resource contention, high availability, breakpoint, …
5. **System Architecture-based –** Focused on a tier or component
6. **Interactions –** Contention between parallel systems / events
   - Rendezvous, Interference, Deadlock, synchronization…

Test Data Pattern Categories – cont.

7. **Human Error –** Focus on worst-case user scenarios
   - "Bad day", Soap Opera
8. **Catastrophe –** Disaster recovery
   - Simulating actual failure load scenario
9. **Physical Failure**
10. **Handling Changes**
11. **Handling Errors**
12. **Risk-based**

See Patterns.doc for full compendium, descriptions

Example 1: Measurable Behavior/Throughput

- **Application:** Retail Product Catalog linking Buyers & Suppliers
- **Architecture:** 3 channel interface, via EDI, MQ and http
- **Test Objective:** Compare file throughput between legacy DB2/mainframe and new Oracle/Unix systems to mitigate risk of migrating all customer to Oracle
- **Live data:** Harvested from DB2 Prod and Oracle Prod Pilot
- **Data Pattern enhancements:**
  - Environment: File header IDs modified to run in TEST
  - Reusability: DB snapshots taken *prior* to harvesting date so that transactions would process correctly
  - Peak: Selected a “peak volume” week for harvesting data
  - Measureable behavior: File volumes adjusted so that same number of files is sent to both systems concurrently

Example 1 - Results

- **Live data assessment:**
  - Realism: High; data included files with real errors & processed against a “real” DB snapshot (size, state)
  - Volume: Realistic “peak volumes” on both systems
- **Test results:**
  - Throughput comparisons showed Oracle 20% higher than DB2, and with more consistent file response times
- **Conclusion:**
  - Business moved forward with full customer migration to Oracle

Example 2: Peak/ Measurable Behavior/ Response

- **Application:** Mobile phone Ringtone Download Manager
- **Architecture:** Mobile Access Gateway over http to web app, in turn over http to billing server
- **Test Objective:** Determine scalability of Download Manager
- **Live data:** Harvested weblogs from two production systems serving distinct mobile handset communities (CDMA & WCDMA)
- **Data Pattern enhancements:**
  - Tier-specific: By-passed MAG, emulating its http interaction with Download Manager
  - Environment: Billing numbers modified to run in TEST; filtered out requests for handset types for which downloads not applicable
  - Peak: Harvested weblogs from an average week; grew requests & concurrent sessions to simulate peak of 2x
  - Measureable behavior: Replay requests at same arrival rate as in Prod

Example 2 - Results

- **Live data assessment:**
  - Realism: High; replayed actual http requests against Download Manager interacting with copy of Prod Billing DB
  - Volume: Realistic “projected peak volumes” at same arrival rate
- **Test results:**
  - Response time degraded over load, overloading the DM at 300 users
- **Conclusion:**
  - DM vendor needed to diagnose & tune the java middle tier

Benefits of Live Data

- **Reality-based**
  - Capture over time; provides trends
  - “messy richness”
- **Enhancements are context-driven, deliberate**
  - Guided by test objectives
  - Engineered to be reusable
- **Faster time-to-first-test**
  - Faster to capture / extract and enhance than to fabricate
- **Lower cost of data prep**

Limitations of Live Data

- Sometimes requires pre-processing effort to run in test environment
- If system performs acceptably…
  - The same “messy richness” can lull us into declaring success…though we may not understand fully what this data contains
- Catch 22s:
  - If live data had the ability to find a performance problem, we would have seen it in production already (have you?)
  - If fabricated data surfaces a problem, what confidence to we have that this exists in real world?
  - If the system is new there may not be any live data to use

In Conclusion, 3 Simple Questions…

…based on our project goals and context…

1. What live data should we use?
2. How do we capture and enhance this data?
3. How do we use it in testing?

Resources

- Appendix*: Live Data Patterns Compendium*
- Feb'09 STP Mag article *Increase Test Validity with Live Data Loads* introduces live data and compares its effectiveness with 'generated' and 'fabricated' data

Questions?

rcollard@attglobal.net

ddowning@mentora.com